ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs

Yang Liu[1], Xiangji Huang[2], Aijun An[1] and Xiaohui Yu[2]
[1]Department of Computer Science and Engineering
York University, Toronto, Canada
[2]School of Information Technology
York University, Toronto, Canada
yliu@cse.yorku.ca, jhuang@yorku.ca, aan@cse.yorku.ca, xhyu@yorku.ca

ABSTRACT

Due to its high popularity, Weblogs (or blogs in short) present a wealth of information that can be very helpful in assessing the general public's sentiments and opinions. In this paper, we study the problem of mining sentiment information from blogs and investigate ways to use such information for predicting product sales performance. Based on an analysis of the complex nature of sentiments, we propose Sentiment PLSA (S-PLSA), in which a blog entry is viewed as a document generated by a number of hidden sentiment factors. Training an S-PLSA model on the blog data enables us to obtain a succinct summary of the sentiment information embedded in the blogs. We then present ARSA, an autoregressive sentiment-aware model, to utilize the sentiment information captured by S-PLSA for predicting product sales performance. Extensive experiments were conducted on a movie data set. We compare ARSA with alternative models that do not take into account the sentiment information, as well as a model with a different feature selection method. Experiments confirm the effectiveness and superiority of the proposed approach.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Text Mining

General Terms

Algorithm

Keywords

Sentiment mining, blog, autoregressive model

1. INTRODUCTION

In recent years, Weblogs (or blogs in short) have become a popular type of media on the Web. As of November 2006, the blog search engine Technorati was tracking more than 63 million blogs [21]. Blogs are often online diaries published in reverse chronological order, and they can also be commentaries or discussions on a particular subject, ranging from mainstream topics (e.g., food, music, products, politics, etc.), to highly personal interests [13]. Since many bloggers choose to express their opinions online, blogs serve as an excellent indicator of public sentiments and opinions.

This paper studies the predictive power of opinions and sentiments expressed in blogs. We focus on the blogs that contain reviews on products. Since what the general public thinks of a product can no doubt influence how good it sells, understanding the opinions and sentiments expressed in the relevant blogs is of high importance, because these blogs can be a very good indicator of the product's future sales performance. In this paper, we are concerned with developing models and algorithms that can mine opinions and sentiments from blogs and use them for predicting product sales. Properly utilized, such models and algorithms can be highly helpful in various aspects of business intelligence, ranging from market analysis to product planning and targeted advertising.

As a case study, we investigate how to predict box office revenues using the sentiment information obtained from blog mentions. The choice of using movies rather than other products in our study is mainly due to data availability, in that the daily box office revenue data are all published on the Web and readily available, unlike other product sales data which are often private to their respective companies due to obvious reasons. Also, as discussed by Liu et al. [15], analyzing movie reviews is one of the most challenging tasks in sentiment mining. We expect the models and algorithms developed for box office prediction to be easily adapted to handle other types of products that are subject to online discussions, such as books, music CDs and electronics.

Prior studies on the predictive power of blogs have used the volume of blogs or link structures to predict the trend of product sales [7, 8], failing to consider the effect of the sentiments present in the blogs. It has been reported [7, 8] that although there seems to exist strong correlation between the blog mentions and sales spikes, using the volume or the link structures alone do not provide satisfactory prediction performance. Indeed, as we will illustrate with an example, the sentiments expressed in the blogs are more predictive than volumes.

Mining opinions and sentiments from blogs, which is necessary for predicting future product sales, presents unique challenges that can not be easily addressed by conventional text mining methods. Therefore, simply classifying blog reviews as positive and negative, as most current sentiment-mining approaches are designed for, does not provide a com-

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

prehensive understanding of the sentiments reflected in the blog reviews. In order to model the multifaceted nature of sentiments, we view the sentiments embedded in blogs as an outcome of the joint contribution of a number of hidden factors, and propose a novel approach to sentiment mining based on Probabilistic Latent Semantic Analysis (PLSA), which we call Sentiment PLSA (S-PLSA). Different from the traditional PLSA [9], S-PLSA focuses on sentiments rather than topics. Therefore, instead of taking a vanilla "bag of words" approach and considering all the words (modulo stop words) present in the blogs, we focus primarily on the words that are sentiment-related. To this end, we adopt in our study the appraisal words extracted from the lexicon constructed by Whitelaw et al. [24]. Despite the seemingly lower word coverage (compared to using "bag of words"), decent performance has been reported when using appraisal words in sentiment classification [24]. In S-PLSA, appraisal words are exploited to compose the feature vectors for blogs, which are then used to infer the hidden sentiment factors.

Aside from the S-PLSA model which extracts the sentiments from blogs for predicting future product sales, we also consider the past sale performance of the same product as another important factor in predicting the product's future sales performance. We capture this effect through the use of an autoregressive (AR) model, which has been widely used in many time series analysis problems, including stock price prediction [6]. Combining this AR model with sentiment information mined from the blogs, we propose a new model for product sales prediction called the Autoregressive Sentiment Aware (ARSA) model. Extensive experiments on the movie dataset has shown that the ARSA model provides superior predication performance compared to using the AR model alone, confirming our expectation that sentiments play an important role in predicting future sales performance.

In summary, we make the following contributions.

- We are the first to model sentiments in blogs as the joint outcome of some hidden factors, answering the call for a model that can handle the complex nature of sentiments. We propose the S-PLSA model, which through the use of appraisal groups, provides a probabilistic framework to analyze sentiments in blogs.

- We propose the Autoregressive Sentiment Aware (ARSA) model for product sales prediction, which reflects the effects of both sentiments and past sales performance on future sales performance. Its effectiveness is confirmed by experiments.

The rest of the paper is organized as follows. Section 2 provides a brief review of related work. In Section 3, we discuss the characteristics of online discussions and specifically, blogs, which motivate the proposal of S-PLSA in Section 4. In Section 5, we propose ARSA, the sentiment-aware model for predicting future product sales. Section 6 reports on the experimental results. We conclude the paper in Section 7.

2. RELATED WORK

2.1 Sentiment Mining

Most existing work on sentiment mining (sometimes also under the umbrella of opinion mining) focuses on determining the semantic orientations of documents. Among them, some of the studies attempt to learn a positive/negative classifier at the document level. Pang et al. [20] employ three machine learning approaches (Naive Bayes, Maximum Entropy, and Support Vector Machine) to label the polarity of IMDB movie reviews. In a follow up work, they propose to firstly extract the subjective portion of text with a graph min-cut algorithm, and then feed them into the sentiment classifier [18]. Instead of applying the straightforward frequency-based bag-of-words feature selection methods, Whitelaw et al. [24] define the concept of "adjectival appraisal groups" headed by an appraising adjective and optionally modified by words like "not" or "very". Each appraisal group is further assigned four type of features: attitude, orientation, graduation, and polarity. They report good classification accuracy using appraisal groups. They also show that when combined with standard "bag-of-words" features, the classification accuracy can be further boosted. We use the same words and phrases from the appraisal words to compute the blogs' feature vectors, as we also believe that such adjectival appraisal words play a vital role in sentiment mining and need to be distinguished from other words. However, as will become evident in Section 4, our way of using these appraisal groups is different from that in [24].

There are also studies that work at a finer level and use words as the classification subject. They classify words into two groups, "good" and "bad", and then use certain functions to estimate the overall "goodness" or "badness" score for the documents. Kamps et al. [11] propose to evaluate the semantic distance from a word to good/bad with WordNet. Turney [23] measures the strength of sentiment by the difference of the mutual information (PMI) between the given phrase and "excellent" and the PMI between the given phrase and "poor".

Pushing further from the explicit two-class classification problem, Pang et al. [19] and Zhang [25] attempt to determine the author's opinion with different rating scales (i.e., the number of stars). Liu et al. [15] build a framework to compare consumer opinions of competing products using multiple feature dimensions. After deducting supervised rules from product reviews, the strength and weakness of the product are visualized with an "Opinion Observer".

Our method departs from classic sentiment classification in that we assume that sentiment consists of multiple hidden aspects, and use a probability model to quantitatively measure the relationship between sentiment aspects and blogs, as well as sentiment aspects and words.

2.2 Blog Mining

Blogs have recently attracted a lot of research interest. There are currently two major research directions on blog analysis. One direction is to make use of links or URLs in Blogspace [22, 1, 12, 5, 7]. Kumar et al. [12] build a time graph for Blogspace, and develop views of the graph as a function of time. By observing the evolving behavior of the time graph, burst defined as a large sequence of temporally focused documents with plenty of links between them can be traced. Efron [5] describes a hyperlink-based method to estimate the political orientation of Web documents. By estimating the likelihood of cocitation between a document of interest and documents with known orientations, the unknown document is classified to either left- or right-wing community. Gruhl et al. [7, 8] prove that there is a strong correlation between blog mentions and sales rank; they therefore believe that the temporal information of topics may help to forecast spike patterns in sales rank.

The other direction focuses on analyzing the contents of blogs [14, 17, 16, 3]. Mei et al. [17] considers blog as a mixture of unigram language models, with each component corresponding to a distinct subtopic or theme. To analyze spatiotemporal theme patterns from blogs, location variable and theme snapshot for each given time period are integrated in the model. Dalli [3] builds a system for large-scale spatiotemporal analysis of online news and blogs. He utilizes an embedded hierarchical geospatial database to distinguish geographical named entities, and provides results for an extremely fine-grained analysis of items in news contents.

3. CHARACTERISTICS OF ONLINE DISCUSSIONS

Intuitively, a newly released product that evokes a lot of online discussions is likely to have an outstanding sales performance. However, evidences show that even if there exists a strong correlation between the number of blog mentions of a new product and the sales rank of the product, it could still be very difficult to make a successful prediction of sales ranks based on the number of blog mentions [7]. To make a better understanding of the characteristics of online discussions and their predictive power, we investigate the pattern of blog mentions and its relationship to sales data by examining a real example from the movie sector.

3.1 Blog mentions

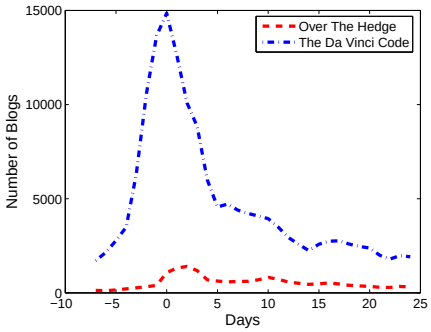
Let us look at the following two movies, *The Da Vinci Code* and *Over the Hedge*, which are both released on May 19, 2006. We use the name of each movie as a query to a publicly available blog search engine [1]. In addition, as each blog is always associated with a fixed time stamp, we augment the query input with a date for which we would like to collect the data. For each movie, by issuing a separate query for each single day in the period starting from one week before the movie release till three weeks after the release, we chronologically collect a set of blogs appearing in a span of one month. We use the number of returned results for a particular date as a rough estimate of the number of blog mentions published on that day.

In Figure 1 (a), we compare the changes in the number of blog mentions of the two movies. Apparently, there exists a spike in the number of blog mentions for the movie *The Da Vinci Code*, which indicates that a large volume of discussions on that movie appeared around its release date. In addition, the number of blog mentions are significantly larger than those for *Over the Hedge* throughout the whole month.

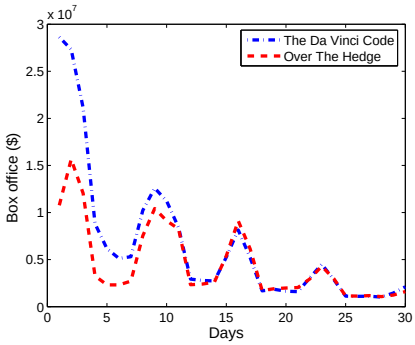
3.2 Box office data and user rating

Besides the blogs, we also collect for each movie one month's box office data (daily gross revenue) from the IMDB website [2]. The changes in daily gross revenues are depicted in Figure 1 (b). Apparently, the daily gross of *The Da Vinci Code* is much greater than *Over the Hedge* on the release date. However, the difference in the gross revenues between the two movies becomes less and less as time goes by, with *Over the Hedge* sometimes even scoring higher towards the end of the one-month period. To shed some light on this phenomenon, we collect the average user ratings of the two movies from the IMDB website. *The Da Vinci Code* and *Over the Hedge* got the rating of 6.5 and 7.1 respectively.

[1] http://www.google.ca/blogsearch?hl=en
[2] http://www.imdb.com/



(a) Change in the number of blogs over time



(b) Change of box office revenues over time

Figure 1: An Example

3.3 Discussion

It is interesting to observe from Figure 1 that although *The Da Vinci Code* has a much higher number of blog mentions than *Over the Hedge*, its box office revenue are on par with that of *Over the Hedge* save the opening week. This implies that the number of blog mentions may not be an accurate indicator of a product's sales performance. A product can attract a lot of attention (thus a large number of blog mentions) due to various reasons, such as aggressive marketing, unique features, or being controversial. This may boost the product's performance for a short period of time. But as time goes by, it is the quality of the product and how people feel about it that dominates. This can partly explain why in the opening week, *The Da Vinci Code* had a large number of blog mentions and staged an outstanding box office performance, but in the remaining weeks, its box office performance fell to the same level as that for *Over the Hedge*. On the other hand, people's opinions (as reflected by the user ratings) seem to be a good indicator of how the box office performance evolves. Observe that, in our example, the average user rating for *Over the Hedge* is higher than that for *The Da Vinci Code*; at the same time, it enjoys a slower rate of decline in box office revenues than the latter. This suggests that sentiments in the blogs could be a very good indicator of a product's future sales performance.

4. S-PLSA: A PROBABILISTIC APPROACH TO SENTIMENT MINING

In this section, we propose a probabilistic approach to analyzing sentiments in the blogs, which will serve as the basis for predicting sales performance.

4.1 Feature Selection

We first consider the problem of feature selection, i.e., how to represent a given blog as an input to the mining algorithms. The traditional way to do this is to compute the (relative) frequencies of various words in a given blog and use the resulting multidimensional feature vector as the representation of the blog. Here we follow the same methodology. But instead of using the frequencies of all the words appearing the blogs, we choose to focus on the set containing 2030 appraisal words extracted from the lexicon constructed by Whitelaw et al. [24], and use their frequencies in a blog as a feature vector. The rationale behind this is that for sentiment analysis, sentiment-oriented words, such as "good" or "bad", are more indicative than other words [24].

4.2 Sentiment PLSA

Mining opinions and sentiments present unique challenges that cannot be handled easily by traditional text mining algorithms. This is mainly because the opinions and sentiments, which are usually written in natural languages, are often expressed in complex ways. Moreover, sentiments are often multifaceted, and can differ from one another in a variety of ways, including polarity, orientation, graduation, and so on. Therefore, it would be too simplistic to just classify the sentiments expressed in a blog as either positive or negative. For the purpose of sales prediction, a model that can extract the sentiments in a more accurate way is needed.

To this end, we propose a probabilistic model called Sentiment Probabilistic Latent Semantic Analysis (S-PLSA), in which a blog can be considered as being generated under the influence of a number of hidden sentiment factors. The use of hidden factors allows us to accommodate the intricate nature of sentiments, with each hidden factor focusing on one specific aspect of the sentiments. The use of a probabilistic generative model, on the other hand, enables us to deal with sentiment analysis in a principled way. In its traditional form, PLSA [9] assumes that there are a set of hidden semantic factors or *aspects* in the documents, and models the relationship among these factors, documents, and words under a probabilistic framework. With its high flexibility and solid statistical foundations, PLSA has been widely used in many areas, including information retrieval, Web usage mining, and collaborative filtering. Nonetheless, to the best of our knowledge, we are the first to model sentiments and opinions as a mixture of hidden factors and use PLSA for sentiment mining.

We now formally present S-PLSA. Suppose we are given a set of blog entries $\mathcal{B} = \{b_1, \ldots, b_N\}$, and a set of words (appraisal words) from a vocabulary $\mathcal{W} = \{w_1, \ldots, w_M\}$. The blog data can be described as a $N \times M$ matrix $D = (c(b_i, w_j))_{ij}$, where $c(b_i, w_j)$ is the number of times w_i appears in blog entry b_j. Each row in D is then a frequency vector that corresponds to a blog entry. We consider the blog entries as being generated from a number of hidden sentiment factors, $\mathcal{Z} = \{z_1, \ldots, z_K\}$. We expect that those hidden factors would correspond to blogger's complex sentiments expressed in the blog review. S-PLSA can be considered as the following generative model.

1. Pick a blog document b from $\mathcal{B}$ with probability $P(b)$;
2. Choose a hidden sentiment factor z from $\mathcal{Z}$ with probability $P(z|b)$;
3. Choose a word from the set of appraisal words $\mathcal{W}$ with probability $P(w|z)$.

The end result of this generative process is a blog-word pair (b, w), with z being integrated out. The joint probability can be factored as follows:

$$P(b, w) = P(b)P(w|b),$$

where

$$P(w|b) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|b).$$

Assuming that the blog entry b and the word w are conditionally independent given the hidden sentiment factor z, we can use Bayes rule to transform the joint probability to the following:

$$P(b, w) = \sum_{z \in \mathcal{Z}} P(z)P(b|z)P(w|z).$$

To explain the observed (b, w) pairs, we need to estimate the model parameters $P(z)$, $P(b|z)$, and $P(w|z)$. To this end, we seek to maximize the following likelihood function:

$$L(\mathcal{B}, \mathcal{W}) = \sum_{b \in \mathcal{B}} \sum_{w \in \mathcal{W}} c(b, w) \log P(b, w),$$

where $c(b, w)$ represents the number of occurrences of a pair (b, w) in the data.

A widely used method to perform maximum likelihood parameter estimation for models involving latent variables (such as our S-PLSA model) is the Expectation-Maximization (EM) algorithm [4], which involves an iterative process with two alternating steps: (1) an expectation step (E-step), where posterior probabilities for the latent variables (in our case, the variable z) are computed, based on the current estimates of the parameters; (2) a maximization step (M-step), where estimates for the parameters are updated to maximize the complete data likelihood.

In our model, with the parameters $P(z)$, $P(w|z)$, and $P(b|z)$ suitably initialized, we can show that the algorithm requires alternating between the following two steps:

- in E-step, we compute

$$P(z|b, w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z' \in \mathcal{Z}} P(z')P(d|z')P(w|z')};$$

- in M-step, we update the model parameters with

$$P(w|z) = \frac{\sum_{b \in \mathcal{B}} c(b, w)P(z|b, w)}{\sum_{b \in \mathcal{B}} \sum_{w' \in \mathcal{W}} c(b, w')P(z|b, w')},$$

$$P(b|z) = \frac{\sum_{w \in \mathcal{W}} c(b, w)P(z|b, w)}{\sum_{b' \in \mathcal{B}} \sum_{w \in \mathcal{W}} c(b', w)P(z|b', w)},$$

$$P(z) = \frac{\sum_{b \in \mathcal{B}} \sum_{w \in \mathcal{W}} c(b, w)P(z|b, w)}{\sum_{b \in \mathcal{B}} \sum_{w \in \mathcal{W}} c(b, w)}.$$

It can be shown that each iteration above monotonically increases the complete data likelihood, and the algorithm converges when a local optimal solution is achieved.

Once the parameter estimation for the model is completed, we can compute the posterior probability $P(z|b)$ using the Bayes rule:

$$P(z|b) = \frac{P(b|z)P(z)}{\sum_{z \in \mathcal{Z}} P(b|z)P(z)}.$$

Intuitively, $P(z|b)$ represents how much a hidden sentiment

factor $z(\in \mathcal{Z})$ "contributes" to the blog document b. Therefore, the set of probabilities $\{P(z|b)|z \in \mathcal{Z}\}$ can be considered as a succinct summarization of b in terms of sentiments. As will be shown in the next section, this summarization can then be used in the predication of future product sales.

5. ARSA: A SENTIMENT-AWARE MODEL

We now present a model to provide product sales predications based on the sentiment information captured from blogs. Due to the complex and dynamic nature of sentiment patterns expressed through on-line chatters, integrating such information is quite challenging. To the best of our knowledge, we are the first to consider using sentiment information for product sales prediction.

We focus on the case of predicting box office revenues to illustrate our methodologies. Our model aims to capture two different factors that can affect the box office revenue of the current day. One factor is the box office revenue of the preceding days. Naturally, the box office revenue of the current day is strongly correlated to those of the preceding days, and how a movie performs in previous days is a very good indicator of how it will perform in the days to come. The second factor we consider is the people's sentiments about the movie. The example in Section 3 shows that a movie's box office is closely related to what people think about the movie. Therefore, we would like to incorporate the sentiments mined from the blogs into the prediction model.

5.1 The autoregressive model

We start with a model that captures only the first factor described above and discuss how to incorporate the second factor into the model in the next subsection.

The temporal relationship between the box office revenues of the preceding days and the current day can be well modeled by an autoregressive (AR) process. Let us denote the box office revenue of the movie of interest at day t by x_t ($t = 1, \ldots, N$ where $t = 1$ corresponds to the release date and $t = N$ corresponds to the last date we are interested in), and we use $\{x_t\}$ to denote the time series $x_1, x_2, \ldots, x_N$. Our goal is to obtain an AR process that can model the time series $\{x_t\}$. A basic (but not quite appropriate, as discussed below) AR process of order p is as follows:

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t,$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters of the model, and ϵ_t is an error term (white noise with zero mean).

Once this model is learned from training data, at day t, the box office revenue x_t can be predicted by x_{t-1}, x_{t-2}, ..., x_{t-p}. It is important to note, however, that AR models are only appropriate for time series that are stationary. Apparently, the time series $\{x_t\}$ are not, because there normally exist clear trends and "seasonalities" in the series. For instance, in example 3, there is a seemingly negative exponential downward trend for the box office revenues as the time moves further from the release date. "Seasonality" is also present, as within each week, the box office revenues always peak at the weekend and are generally lower during weekdays. Therefore, in order to properly model the time series $\{x_t\}$, some preprocessing steps are required.

The first step is to remove the trend. This is achieved by first transforming the time series $\{x_t\}$ into the logarithmic domain, and then differencing the resulting time series $\{x_t\}$. The new time series obtained is thus $x'_t = \Delta \log x_t = \log x_t - \log x_{t-1}$. We then proceed to remove the seasonality [6]. To this end, we apply the lag operator on $\{x'_t\}$ and obtain a new time series $\{y_t\}$ as follows:

$$y_t = x'_t - L^7 x' t = x'_t - x'_{t-7}.$$

By computing the difference between the box office revenue of a particular date and that of 7 days ago, we effectively removed the seasonality factor due to different days of a week. After the preprocessing step, a new AR model can be formed on the resulting time series $\{y_t\}$:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t. \quad (1)$$

It is worth noting that although the AR model developed here is specific for movies, the same methodologies can be applied in other contexts. For example, trends and seasonalities are present in the sales performance of many different products (such as electronics and music CDs). Therefore the preprocessing steps described above to remove them can be adapted and used in the predicting the sales performance.

5.2 Incorporating sentiments

As discussed earlier, the box office revenues might be greatly influenced by people's opinions in the same time period. We modify the model in (1) to take this factor into account. Let $\mathcal{B}_t$ denote the set of blogs on the movie of interest that were posted on day t. The average probability of sentiment factor $z = j$ conditional on blogs in $\mathcal{B}_t$ is defined as

$$\omega_{t,j} = \frac{1}{|\mathcal{B}_t|} \sum_{b \in \mathcal{B}_t} p(z = j|b),$$

where $p(z = j|b)(b \in \mathcal{B}_t)$ are obtained based a trained S-PLSA model. Intuitively, $\omega_{t,j}$ represents the average fraction of the sentiment "mass" that can be attributed to the hidden sentiment factor j. Then our new model, which we call the Autoregressive Sentiment-Aware (ARSA) model, can be formulated as follows.

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \sum_{j=1}^{K} \rho_{i,j} \omega_{t-i,j} + \epsilon_t, \quad (2)$$

where p, q, and K are user-chosen parameters, while ϕ_i and $\rho_{i,j}$ are parameters whose values are to be estimated using the training data. Parameter q specifies the sentiment information from how many preceding days are considered, and K indicates the number of hidden sentiment factors used by S-PLSA to represent the sentiment information.

In summary, the ARSA model mainly comprises two components. The first component, which corresponds to the first term in the right hand side of Equation (2), reflects the influence of past box office revenues. The second component, which corresponds to the second term, represents the effect of the sentiments as reflected from the blogs.

5.3 Training the ARSA model

Training the ARSA model involves learning the set of parameters $\phi_i (i = 1, \ldots, p)$, and $\rho_{i,j} (i = 1, \ldots, q; j = 1, \ldots, K)$, from the training data that consist of the true box office revenues, and $\omega_{t,j}$ obtained from the blog data. As we will show below, the model can, after choosing p and q, be fitted by least squares regression to estimate the parameter values.

For a particular movie m ($m = 1, \ldots, M$), where M is the total number of movies in the training data, and a given

date t, let us add the subscript m to y_t and $\omega_{t-i,j}$ in Equation (2) to be more precise. Let $\alpha_{m,t} = (y_{m,t-1}, \ldots, y_{m,t-p}, \omega_{m,t-1,1}, \ldots, \omega_{m,t-q,k})^T$. Then Equation (2) can be rewritten as $\alpha_{m,t}^T \theta = y_{m,t}$, Let A be a matrix composed of all $\alpha_{m,t}$ vectors corresponding to each movie and, for each movie and each t, i.e., $A = (\alpha_{1,1}, \alpha_{1,2}, \ldots)^T$. Similarly, let $\mathbf{c}$ denote the vector consisting of all possible $y_{m,t}$, i.e., $\mathbf{c} = (y_{1,1}, y_{1,2}, \ldots)$. Then based on the training data, we seek to find a solution $\hat{\theta}$ for the "equation"

$$A\theta \approx \mathbf{c}.$$

More precisely, we seek to minimize the Euclidean norm squared of residual $A\theta - \mathbf{c}$. This is exactly a least squares regression problem, and can be solved using standard techniques in mathematics.

Once the model is trained, Equation (2) can be used to predict the box office revenue of day t based on the box office revenues of the preceding days (which have already observed before day t), and the sentiments mined from the blogs.

6. EMPIRICAL STUDY

In this section, we report the results obtained from a set of experiments conducted on a movie data set in order to validate the effectiveness of the proposed model, and compare it against alternative methods.

6.1 Experiment settings

The movie data we used in the experiments consists of two components. The first component is a set of blog documents on movies of interest collected from the Web, and the second component contains the corresponding daily box office revenue data for these movies.

Blog entries were collected for movies released in the United States during the period from May 1, 2006 to August 8, 2006. For each movie, using the movie name and a date as keywords, we composed and submitted queries to Google's blog search engine, and retrieved the blogs entries that were listed in the query results. For a particular movie, we only collected blog entries that had a timestamp ranging from one week before the release to four weeks after, as we assume that most of the reviews might be published close the release date. Through limiting the time span for which we collect the data, we are able to focus on the most interesting period of time around a movie's release, during which the blog discussions are generally the most intense. As a result, the amount of blog entries collected for each movie ranges from 663 (for *Waist Deep*) to 2069 (for *Little Man*). In total, 45046 blog entries that comment on 30 different movies were collected. We then extracted the title, permlink, free text contents, and time stamp from each blog entry, and indexed them using Apache Lucene[3].

We manually collected the gross box office revenue data for the 30 movies from the IMDB website[4]. For each movie, we collected its daily gross revenues in the US starting from the release date till four weeks after the release. In each run of the experiment, the following procedure was followed:

1. We randomly choose half of the movies for training, and the other half for testing; the blog entries and box office revenue data are correspondingly partitioned into training and testing data sets.
2. Using the training blog entries, we train an S-PLSA model. For each blog entry b, the sentiments towards a movie are summarized using a vector of the posterior probabilities of the hidden sentiment factors, $P(z|b)$.
3. We feed the probability vectors obtained in step 2, along with the box revenues of the preceding days, into the ARSA model, and obtain estimates of the parameters.
4. We evaluate the prediction performance of the ARSA model by experimenting it with the testing data set.

In this paper, we use the *mean absolute percentage error* (MAPE) [10] to measure the prediction accuracy:

$$MAPE = \frac{1}{n} \sum_{i=1}^{N} \frac{|Pred_i - True_i|}{True_i},$$

where n is the total amount of predictions made on the testing data, $Pred_i$ is the predicted value, and $True_i$ represents the true value of the box office revenue. All the accuracy results reported herein are averages of 30 independent runs.

6.2 Parameter selection

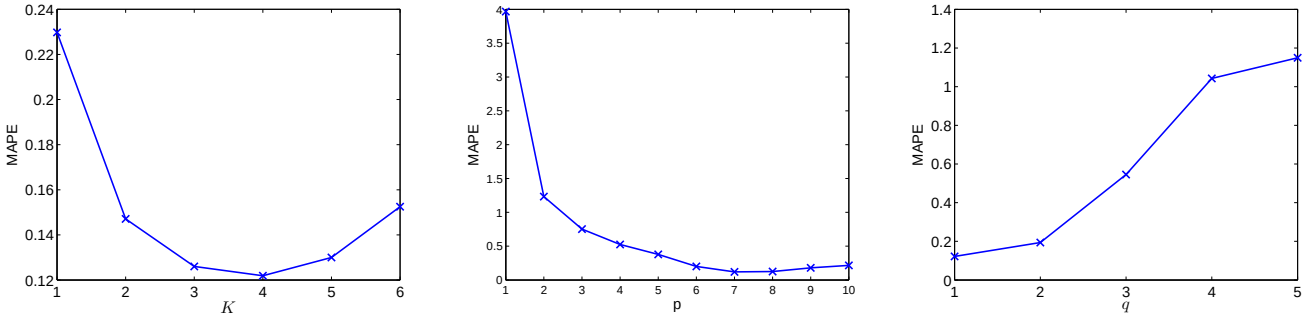
In the ARSA model, there are several user-chosen parameters that provide the flexibility to fine tune the model for optimal performance. They include the number of hidden sentiment factors in S-PLSA, K, and the orders of the ARSA model, p and q. We now study how the choice of these parameter values affects the prediction accuracy.

We first vary K, with fixed p and q values ($p = 7$, and $q = 1$). As shown in Figure 2 (a), as K increases from 1 to 4, the prediction accuracy improves, and at $K = 4$, $ARSA$ achieves an MAPE of 12.1%. That implies that representing the sentiments with higher dimensional probability vectors allows S-PLSA to more fully capture the sentiment information, which leads to more accurate prediction. On the other hand, as shown in the graph, the prediction accuracy deteriorates once K gets past 4. The explanation here is that a large K may cause the problem of overfitting [2], i.e., the S-PLSA might fit the training data better with a large K, but its generalization capability on the testing data might become poor. Some tempering algorithms have been proposed to solve the overfitting problem [9], but it is out of the scope of our study. Also, if the number of appraisal words used to train the model is M, and the number of blog entries is N, the total number of parameters which must be estimated in the S-PLSA model is $K(M + N + 1)$. This number grows linearly with respect to the number of hidden factors K. If K gets too large, it may incur a high training cost in terms of time and space.

We then vary the value of p, with $K = 4$ and $q = 1$ to study how the order of the autoregressive model affects the prediction accuracy. We observe from Figure 2 (b) that the model achieves it best prediction accuracy when $p = 7$. This suggests that p should be large enough to factor in all the significant influence of the preceding days' box office performance, but not too large to let irrelevant information in the more distant past to affect the prediction accuracy.

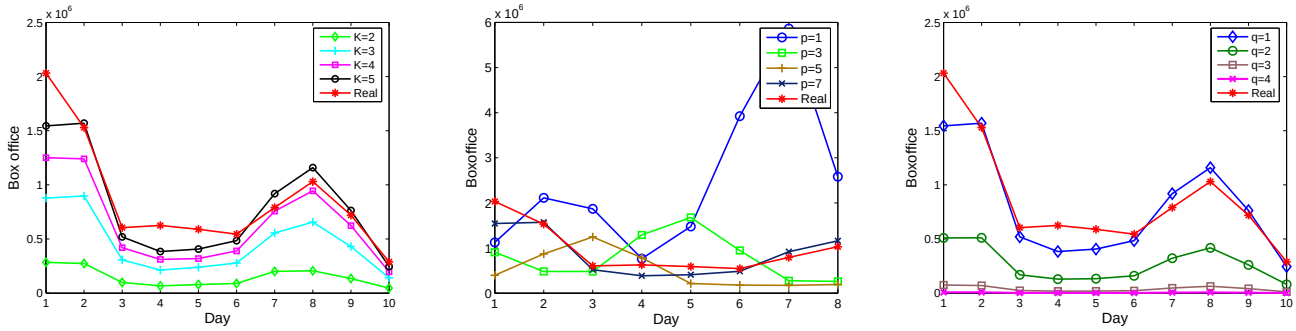
Using the optimal values of K and p, we vary q from 1 to 5 to study its effect on the prediction accuracy. As shown in Figure 2 (c), the best prediction accuracy is achieved at $q = 1$, which implies that the prediction is most strongly related to the sentiment information captured from blog entries posted on the immediately preceding day.

[3] `http://lucene.apache.org`
[4] `http://www.imdb.com`



(a) Effect of K (b) Effect of p (c) Effect of q

Figure 2: The effects of parameters on the prediction accuracy



(a) Effect of K (b) Effect of p (c) Effect of q

Figure 3: The effects of parameters for the movie *Little Man*

To better illustrate the effects of the parameter values on the prediction accuracy, we present in Figure 3 the experimental results on a particular movie, *Little Man*. For each parameter, we plot the predicted box office revenues and the true values for each day using different values of the parameter. It is evident from the plots that the responses to each parameter are similar to what is observed from Figure 2. Also note that the predicted values using the optimal parameter settings are close to the true values accross the whole time span. Similar results are also observed on other movies, demonstrating the consistency of the proposed approach for different days.

6.3 Comparison with alternative methods

To verify that the sentiment information captured by the S-PLSA model plays an important role in box office revenue prediction, we compare ARSA with two alternative methods which do not take sentiment information into consideration.

We first conduct experiments to compare ARSA against the pure autoregressive (AR) model without any terms on sentiments, i.e., $y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$. The results are shown in Figure 4. We observe the behaviors of the two models as p ranges from 3 to 7. Apparently, although the accuracy of both methods improves with increasing p, ARSA constantly outperforms the AR model by a factor of 2 to 3.

We then proceed to compare ARSA with an autoregressive model that factors in the volume of blog mentions in prediction. In Section 3, we have illustrated the characteristics of the volume of blog mentions and its connection to the sales performance with an example, showing that although there exists a correlation between the volume of blog mentions and the sales performance, this correlation may not be strong enough to enable prediction. To further demonstrate

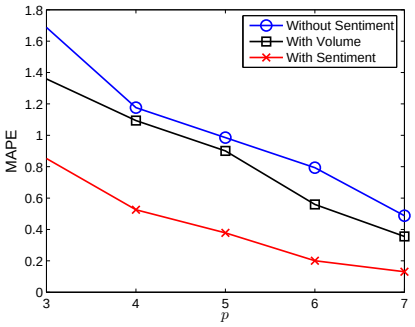


Figure 4: ARSA vs. alternative methods

this, we experiment with the following autoregressive model that utilizes the volume of blogs mentions. In contrast to ARSA, where we use a multi-dimensional probability vector produced by S-PLSA to represent bloggers' sentiments, this model uses a scalar (number of blog mentions) to indicate the degree of popularity. The model can be formulated as

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \rho_i v_{t-i} + \epsilon_t,$$

where y_t's are obtained in the same way as in ARSA, v_{t-i} denotes the number of blog mentions on day $t-i$, and ϕ_i and ρ_i are parameters to be learned. This model can be trained using a procedure similar to what is used for ARSA. Using the same training and testing data sets as what are used for ARSA, we test the performance of this model and compare it with ARSA. The results are shown in Figure 4. Observe that although this method yields a moderate performance gain over the pure AR model (which proves that the volume

data do have some predictive power), its performance is still dominated by the ARSA model.

6.4 Comparison with other feature selection methods

To test the effectiveness of using appraisal words as the feature set, we experimentally compare ARSA with a model that uses the classic bag-of-words method for feature selection, where the feature vectors are computed using the (relative) frequencies of all the words appearing in the blog entries. That is, instead of using the appraisal words, we train an S-PLSA model with the bag-of-words feature set, and feed the probabilities over the hidden factors thus obtained into the ARSA model for training and prediction. Note that, in practice, it is generally infeasible to consider all the words appearing in the blog entries as potential features, because the feature set would be extremely large (in the order of 100,000 in our data set), and the cost of constructing a document-feature matrix could be prohibitively high. To alleviate this problem, only words with higher frequencies (excluding stop words) are selected into the feature set. To ensure a fair comparison, the number of words selected (2,015) is the same as the number of appraisal words used in ARSA. Using $p = 7$ and $q = 1$, we vary K from 2 to 5 and compare the performances of both feature selection methods. As shown in Figure 5, using appraisal words significantly outperforms the bag-of-words approach. Similar trends can be observed when other values of the parameters p, q, and K are used.

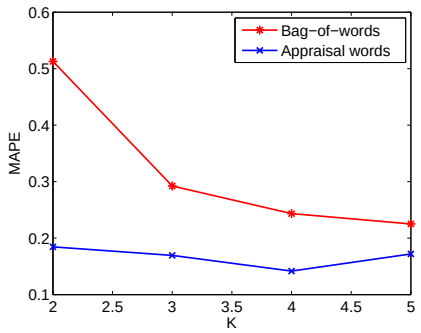


Figure 5: Comparison with bag-of-words

7. CONCLUSIONS AND FUTURE WORK

The wide spread use of blogs as a way of conveying personal views and comments has offered an unique opportunity to understand the general public's sentiments and use this information to advance business intelligence. In this paper, we have explored the predictive power of blogs using movies as a case study, and studied the problem of predicting sales performance using sentiment information mined form blogs. A center piece of our work is the proposal of S-PLSA, a generative model for sentiment analysis that helps us move from simple "negative or positive" classification towards a deeper comprehension of the sentiments in blogs. Using S-PLSA as a means of "summarizing" sentiment information from blogs, we develop ARSA, a model for predicting sales performance based on the sentiment information and the product's past sales performance. The accuracy and effectiveness of our model have been confirmed by the experiments on the movie data set. Equipped with the proposed models, companies will be able to better harness the predictive power of blogs and conduct businesses in a more effective way.

It is worth noting that although we have only used S-PLSA for the purpose of prediction in this work, it is indeed a model general enough to be applied to other scenarios. For future work, we would like to explore its role in clustering and classification of blogs based on their sentiments. Another possible direction for future work is to use S-PLSA as a tool to help track and monitor the changes and trends in sentiments expressed online.

8. REFERENCES

[1] J. Bar-Ilan. An outsider's view on "topic-oriented blogging". In *WWW Alt. '04*, pages 28–34, 2004.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[3] Angelo Dalli. System for spatio-temporal analysis of online news and blogs. In *WWW '06*, pages 929–930, 2006.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *em* algorithm. *Journal of Royal Statistical Society*, B(39):1–38, 1977.

[5] Miles Efron. The liberal media and right-wing conspiracies: using cocitation information to estimate political orientation in web documents. In *CIKM '04*, pages 390–398, 2004.

[6] Walter Enders. *Applied Econometric Time Series*. Wiley, New York, 2nd edition, 2004.

[7] Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The predictive power of online chatter. In *KDD '05*, pages 78–87, 2005.

[8] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *WWW '04*, pages 491–501, 2004.

[9] Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI'99*, 1999.

[10] Wolfgang Jank, Galit Shmueli, and Shanshan Wang. Dynamic, real-time forecasting of online auctions via functional models. In *KDD '06*, pages 580–585, 2006.

[11] Jaap Kamps and Maarten Marx. Words with attitude. In *Proc. of the First International Conference on Global WordNet*, pages 332–341, 2002.

[12] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *WWW '03*, pages 568–576, 2003.

[13] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Structure and evolution of blogspace. *Commun. ACM*, 47(12):35–39, 2004.

[14] Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. A probabilistic model for retrospective news event detection. In *SIGIR '05*, pages 106–113, 2005.

[15] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05*, pages 342–351, 2005.

[16] Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW '06*, pages 533–542, 2006.

[17] Qiaozhu Mei and ChengXiang Zhai. A mixture model for contextual text mining. In *KDD '06*, pages 649–655, 2006.

[18] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04*, pages 271–278, 2004.

[19] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL '05*, pages 115–124, 2005.

[20] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[21] Technorati. URL:`http://technorati.com/about/`. Retrieved on January 27, 2007.

[22] B. L. Tseng, J. Tatemura, and Y. Wu. Tomographic clustering to visualize blog communities as mountain views. In *Proc. of 2nd Annual Workshop on the Weblogging Ecosystem*, 2005.

[23] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02*, pages 417–424, 2001.

[24] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. In *CIKM '05*, pages 625–631, 2005.

[25] Zhu Zhang and Balaji Varadarajan. Utility scoring of product reviews. In *CIKM '06*, pages 51–57, 2006.